Proceedings of the 36th
Conference on Decision & Control
San Diego, California USA • December 1997

Stereo Disparity and L^1 Minimization

Arun Kumar
Department of Aerospace Eng.
University of Minnesota
Minneapolis, MN 55455

Steven Haker
Department of Mathematics
University of Minnesota
Minneapolis, MN 55455

Curt Vogel
Department of Mathematical Sciences
Montana State University
Bozeman, Montana 59717-0240

Allen Tannenbaum
Department of Electrical Engineering
University of Minnesota
Minneapolis, MN 55455

Steven Zucker
Department of Electrical Engineering
Yale University
New Haven, Connecticut 06520

Abstract

In this note, we employ an L^1 pde approach for stereo disparity. This approach has the nice feature of preserving edges in the computation. Our numerical method is based on the techniques for the analysis of curves evolving according to functions of curvature.

1 Introduction

In this note, we consider the problem of finding the depth of objects from a pair of images of a scene taken by two cameras which are laterally displaced from each other. We will assume that most points are visible from the two cameras and so corresponding pixels can be found for them in the two images. A common approach taken for solving this problem is that of *feature matching*. First, corresponding features are selected in the two images. The disparity between matched features can be used along with information about the imaging geometry to compute the actual depth of the matched features. The depth map found in this way will be sparse and the construction of the complete depth map relies on interpolation of the depth values over the remaining points in the image.

The problem of finding matches, whether in low level syntactic features or in higher level features is the most difficult part of the problem. Multiple matches could be found or no match might exist in the case of occlusions. Moreover, the effect of *noise* in giving false matches is quite prominent when low level features are sought to be matched.

In this work we will present an approach to the computation of a dense disparity map using pixel matching. The general idea is to find corresponding pixels in the two images. Then the disparity between the matched pixels along with information about the imaging geometry gives the depth of the point in the scene from the viewer. We will restrict ourselves to matching pixels along horizontal epipolar lines (discussed below). However, the approach presented here can be generalized to matching pixels in the other coordinate directions as well. The approach is based on a constrained optimization problem, but instead of using a quadratic norm as is suggested in [9], we instead choose to take an L^1 norm which has the advantage of preserving edge information better. This type of variational approach was introduced in Rudin *et al.* [16] for image denoising, and employed in [11] for the computation of optical flow.

The literature on stereo matching algorithms in computer vision is vast, and beyond the scope of this note to review here (see, e.g. [2, 6, 10] and the references therein). However, there is one curious aspect of human correspondence perception — the *disparity gradient limit* — that has a fundamental connection to our approach. If one considers disparity as a scalar field in (x, y), the limit with which we are concerned applies to the horizontal component of the gradient of this field. There is clearly a geometrical limit to this based on the spacing between the "eyes" and surface slant. For excessive values, the surface will self-occlude itself from one eye. But human stereo perception appears to obey a limit significantly less than this - for a discussion, see Howard and Rogers [10], Sec. 6.2.7. Disparity gradient limits will emerge implicitly within our algorithm. The only other algorithm that uses a disparity gradient limit of which we are aware is Pollard *et al* [15], but

0-7803-3970-8/97 $10.00 © 1997 IEEE

they impose it as a hard constraint. As we show, this is not necessary, as implied by the psychophysics.

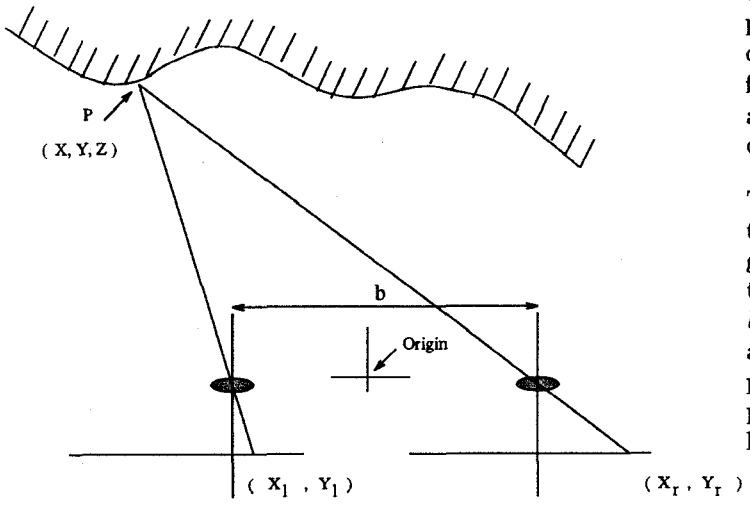

Figure 1: Stereo Geometry

Let us use Figure 1 to describe the imaging geometry. We consider a scene being imaged by two cameras whose optical axes are parallel to each other and perpendicular to the image plane (the image plane is the plane in the camera where the image is recorded on film). The cameras are placed with a distance b between them. The coordinates of a point P (x, y, z) are measured relative to an origin midway between the lens centers. Let us denote the image coordinates of the point in the left and the right image by $(x^{(l)}, y^{(l)})$ and $(x^{(r)}, y^{(r)})$. Then

$$\frac{x^{(l)}}{f} = \frac{x - b/2}{z} \quad \text{and} \quad \frac{x^{(r)}}{f} = \frac{x + b/2}{z}. \tag{1}$$

Also

$$\frac{y^{(l)}}{f} = \frac{y^{(r)}}{f} = \frac{y}{z}. \tag{2}$$

The focal lengths of the two lens are considered equal to f. We would like to solve these equations for the real world coordinates (x, y, z). Note that

$$\frac{x^{(r)} - x^{(l)}}{f} = \frac{b}{z}. \tag{3}$$

The difference $x^{(r)} - x^{(l)}$ is called the *disparity* and we shall denote it with the function $d(x, y)$ (later on we shall solve for two disparity functions: one based on the right image and the other on the left image). Then it is easy to see that the solutions for the coordinates are:

$$x = b\frac{(x^{(l)} + x^{(r)})/2}{d}, \; y = b\frac{(y^{(l)} + y^{(r)})/2}{d}, \; z = b\frac{f}{d}. \tag{4}$$

Thus it is sufficient to solve for the appropriate disparity function since the depth of the objects can be found from equation (4). Also notice that the depth of objects is inversely proportional to the disparity. So objects that are farther will have less disparity between their pixel locations in the left and the right images than closer objects. Depth is directly proportional to the effective focal length f because the images are magnified as the focal length is increased and correspondingly the disparity between pixels would change.

The pair of points that correspond to a certain point in the 3-D world are called the *conjugate pair*. A conjugate pair has to lie on a line since the y coordinates of the conjugate pair are the same. This line is the *epipolar line*. Epipolar lines in the investigation we conduct are all horizontal and so the search for the matching pixel is done only on horizontal lines. This makes the problem simpler but it can be generalized to epipolar lines in arbitrary directions.

2 Pixel Matching

In the pixel matching approach we would like to find the following match: given a pixel in the left image (or in the right image) which is the pixel that corresponds to it in the right image (or in the left image)? Consider three possible "pixel matching terms" $\mathbf{F}^{(l)}(\mathbf{d})(x, y)$, $\mathbf{F}^{(r)}(\mathbf{d})(x, y)$, and $\mathbf{F}^{(c)}(\mathbf{d})(x, y)$, given in terms of a disparity function $\mathbf{d} = \mathbf{d}(x, y)$:

$$\mathbf{F}^{(l)}(\mathbf{d})(x, y) = \mathbf{I}^{(l)}(x, y) - \mathbf{I}^{(r)}(x + \mathbf{d}(x, y), y), \tag{5}$$

$$\mathbf{F}^{(r)}(\mathbf{d})(x, y) = \mathbf{I}^{(l)}(x - \mathbf{d}(x, y), y) - \mathbf{I}^{(r)}(x, y), \tag{6}$$

$$\mathbf{F}^{(c)}(\mathbf{d})(x, y) = \mathbf{I}^{(l)}(x - \frac{1}{2}\mathbf{d}(x, y), y) - \mathbf{I}^{(r)}(x + \frac{1}{2}\mathbf{d}(x, y), y). \tag{7}$$

A solution $\mathbf{d}$ of $\mathbf{F}^{(l)}(\mathbf{d})(x, y) \equiv 0$ is a "left disparity function." It describes the disparity between the position of a pixel in the left image and the corresponding pixel in the right image. Similarly, a solution of $\mathbf{F}^{(r)}(\mathbf{d})(x, y) \equiv 0$ is a "right disparity function". A solution $\mathbf{d}$ of $\mathbf{F}^{(c)}(\mathbf{d})(x, y) \equiv 0$ matches the left and right images by shifting each by half the disparity.

Let $\mathbf{F}(\mathbf{d})(x, y)$ denote one of the three functions $\mathbf{F}^{(l)}(\mathbf{d})(x, y)$, $\mathbf{F}^{(r)}(\mathbf{d})(x, y)$ or $\mathbf{F}^{(c)}(\mathbf{d})(x, y)$. We would like to minimize the following least-squares fit-to-data functional to find the best matching function (disparity function) for corresponding pixels:

$$f_{ls}(\mathbf{d}) = \frac{1}{2}\int\int_{\Omega} [\mathbf{F}(\mathbf{d})(x, y)]^2 \, d\Omega. \tag{8}$$

The (squared) L^2 norm is used primarily for computa-

tional convenience. Unfortunately, the problem of determining the minimizing function $\mathbf{d}$ is ill-posed; small perturbations in image intensity (e.g. due to noise) may yield large purturbations in $\mathbf{d}$. Thus some regularity constraints are necessary to compute solutions. This is very similar to the optical flow problem which is also ill-posed, and our approach for solving it in [11].

Accordingly, we will consider a modified L^1 norm of $\|\nabla \mathbf{d}(x,y)\|$ as the regularizing criterion. In [11], we argue that the L^1 norm minimization is better than a quadratic minimization with respect to edge preserving properties. We hope to gain the same advantage in the computation of depth maps which might have discontinuities at the boundaries of objects. Thus we follow Horn [9] and employ a regularized least squares cost functional

$$f(\mathbf{d}) = \alpha^2 f_{ls}(\mathbf{d}) + f_\beta(\mathbf{d}). \tag{9}$$

where f_β is a regularization, or stabilization, functional

$$f_\beta(\mathbf{d}) = \int\int_\Omega \sqrt{\|\nabla \mathbf{d}(x,y)\|^2 + \beta^2}\, d\Omega \tag{10}$$

of total variation type [8]. Here β is a small positive parameter. When $\beta = 0$, this reduces to the total variation of $\mathbf{d}$, i.e. the L^1 norm of $\|\nabla \mathbf{d}(x,y)\|$. Taking $\beta > 0$ eliminates the difficulty posed by nondifferentiability of the square root when the gradient of $\mathbf{d}$ vanishes. For any $\beta \geq 0$, there is a variational form which extends (10) to allow certain discontinuities in $\mathbf{d}$, e.g., a jump discontinuity along a curve. Total variation type regularization tends to favor "blocky" as opposed to "stringy" disparity functions. See [5]. The parameter α controls the trade-off between fit-to-data and stability. For the purpose of completeness we carry out the computations necessary to find the Euler-Lagrange equations. Define $\mathbf{G}$ to be the following:

$$\mathbf{G} = \frac{1}{2}\alpha^2[\mathbf{F}(\mathbf{d})]^2 + \sqrt{\|\nabla \mathbf{d}\|^2 + \beta^2}. \tag{11}$$

The result of setting the first variation of the cost-functional (9) to zero is:

$$\mathbf{G_d} - \frac{\partial}{\partial x}\mathbf{G_{d_x}} - \frac{\partial}{\partial y}\mathbf{G_{d_y}} = 0. \tag{12}$$

The various terms in the above equation are computed here:

$$\mathbf{G_d} = (\alpha^2 \mathbf{F}(\mathbf{d}))\frac{\partial \mathbf{F}}{\partial \mathbf{d}}(\mathbf{d}) \tag{13}$$

Where $\frac{\partial \mathbf{F}}{\partial \mathbf{d}}(\mathbf{d})$ depends upon the particular type of matching function used:

$$\begin{aligned}
\frac{\partial \mathbf{F}^{(l)}}{\partial \mathbf{d}}(\mathbf{d})(x,y) &= -\frac{\partial \mathbf{I}^{(r)}}{\partial x}(x + \mathbf{d}(x,y), y) \\
\frac{\partial \mathbf{F}^{(r)}}{\partial \mathbf{d}}(\mathbf{d})(x,y) &= -\frac{\partial \mathbf{I}^{(l)}}{\partial x}(x - \mathbf{d}(x,y), y)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathbf{F}^{(c)}}{\partial \mathbf{d}}(\mathbf{d})(x,y) &= -\frac{1}{2}\Big[\frac{\partial \mathbf{I}^{(l)}}{\partial x}(x - \frac{1}{2}\mathbf{d}(x,y), y) \\
&+ \frac{\partial \mathbf{I}^{(r)}}{\partial x}(x + \frac{1}{2}\mathbf{d}(x,y), y)\Big].
\end{aligned}$$

We have

$$\mathbf{G_{d_x}} = \frac{\mathbf{d}_x}{\sqrt{\|\nabla \mathbf{d}(x,y)\|^2 + \beta^2}} \text{ and } \mathbf{G_{d_y}} = \frac{\mathbf{d}_y}{\sqrt{\|\nabla \mathbf{d}(x,y)\|^2 + \beta^2}} \tag{14}$$

and so

$$-\frac{\partial}{\partial x}\mathbf{G_{d_x}} - \frac{\partial}{\partial y}\mathbf{G_{d_y}} = -\nabla \cdot \left(\frac{\mathbf{d}}{\sqrt{\|\nabla \mathbf{d}\|^2 + \beta^2}}\right) =: \mathbf{L}(\mathbf{d})\mathbf{d}. \tag{15}$$

Note that $\mathbf{L}(\mathbf{d})$ is a diffusion operator with disparity-dependent diffusion coefficient $1/\sqrt{\|\nabla \mathbf{d}\|^2 + \beta^2}$. See [12] for details. Of particular interest is the case $\beta = 0$ when we have

$$\frac{\partial \mathbf{G_{d_x}}}{\partial x} + \frac{\partial \mathbf{G_{d_y}}}{\partial y} = \frac{(\mathbf{d}_x)^2 \mathbf{d}_{yy} - 2\mathbf{d}_x \mathbf{d}_y \mathbf{d}_{xy} + (\mathbf{d}_y)^2 \mathbf{d}_{xx}}{((\mathbf{d}_x)^2 + (\mathbf{d}_y)^2)^{3/2}} = \kappa_{\mathbf{d}}, \tag{16}$$

$\kappa_{\mathbf{d}}$ being the curvature of the level sets of the graph of the function $\mathbf{d}$. In this case the Euler-Lagrange equation

$$(\alpha^2 \mathbf{F}(\mathbf{d}))\frac{\partial \mathbf{F}}{\partial \mathbf{d}}(\mathbf{d}) - \nabla \cdot \left(\frac{\mathbf{d}}{\sqrt{\|\nabla \mathbf{d}\|^2 + \beta^2}}\right) = 0 \tag{17}$$

reduces to

$$(\alpha^2 \mathbf{F}(\mathbf{d}))\frac{\partial \mathbf{F}}{\partial \mathbf{d}}(\mathbf{d}) - \kappa_{\mathbf{d}} = 0. \tag{18}$$

3 Numerical Implementation on Test Images

3.1 Numerical scheme

In practice, we are given discrete data consisting of pixel intensities on a rectangular grid for the left and right images, which we denote by

$$\mathbf{I}^{(l)}(x_i, y_j), \ \mathbf{I}^{(r)}(x_i, y_j), \quad 1 \leq i \leq n_x, \ 1 \leq j \leq n_y. \tag{19}$$

The points (x_i, y_j) correspond to pixel, or cell, centers. In place of the cost functional (9), we consider the discrete analogue where the continuous least squares term (8) is replaced by a discrete sum of squares.

Standard gradient-based approaches to minimizing this cost functional require evaluation of terms such as

$$\mathbf{I}^{(l)}\left(x_i - \frac{1}{2}\mathbf{d}(x_i, y_j), y_j\right) \text{ and } \frac{\partial \mathbf{I}^{(l)}}{\partial x}\left(x_i - \frac{1}{2}\mathbf{d}(x_i, y_j), y_j\right), \tag{20}$$

as well as analogous expressions for $\mathbf{I}^{(r)}$, depending upon which form ((5)-(7)) is chosen for $\mathbf{F}(\mathbf{d})(x,y)$.

Since the data is discrete and the $x_i - \frac{1}{2}\mathbf{d}(x_i, y_j)$ need not fall on grid boundaries, some sort of interpolation is required. To evaluate the partial derivatives, this interpolation must be smooth (Note that these requirements may be relaxed if one uses nonstandard, e.g., nonsmooth, optimization techniques). An additional difficulty comes from the fact that the least squares term in the cost functional (9) may be nonconvex. This nonconvexity may occur on a fine scale due to noise in the data. A regularization term like (10) helps to overcome fine-scale nonconvexity. Unfortunately, nonconvexity may also arise on a coarse scale, and the minimization scheme may yield a local, rather than a global, minimizer.

If the cost functional is not smooth and one tries to apply a standard gradient-based minimization scheme, one typically sees nonconvergence due to "cycling". Indeed, an initial attempt was made to solve the Euler-Lagrange equation (18), with $\mathbf{F}(\mathbf{d})(x,y) = \mathbf{F}^{(r)}(\mathbf{d})(x,y)$, using a gradient-descent approach and linear interpolation of $\mathbf{I}^{(l)}$ and $\frac{\partial \mathbf{I}^{(l)}}{\partial x}$. A fictitious time parameter t was introduced, and $\mathbf{d}$ was evolved according to

$$\frac{\partial \mathbf{d}}{\partial t} = \kappa_{\mathbf{d}} - (\alpha^2 \mathbf{F}(\mathbf{d}))\frac{\partial \mathbf{F}}{\partial \mathbf{d}}(\mathbf{d}) \qquad (21)$$

with the hope that a steady state would be achieved. Although there was in general an initial pointwise decrease in the magnitude of the right hand side of (21), it displayed a "sawtooth" behavior and did not tend to zero.

To deal simultaneously with interpolation and smoothing of the data as well as the computation derivatives, the data was mollified, i.e., convolved with a Gaussian kernel. 1-D FFT's can be used to evaluate (smoothed versions of) (20). In particular, for each fixed value of y, let $\mathbf{I}^y(x) := \mathbf{I}^{(l)}(x,y)$ and consider its truncated Fourier series approximation

$$\mathbf{I}^y(x) \approx \sum_{j=-n_x}^{n_x} \hat{\mathbf{I}}_j^y \, e^{j2\pi i x}, \quad i = \sqrt{-1}.$$

Here the $\hat{\mathbf{I}}_j^y$ denote the coefficients of the discrete Fourier transform of the vector $\mathbf{I}^y(x_j)$, $j = -n_x, \ldots, n_x$, extended evenly. Let $\hat{k}_j$ denote the Fourier coefficients of the discrete Gaussian

$$k(x_j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x_j^2/\sigma^2}.$$

The parameter σ determines the width of the Gaussian, and hence, the amount of smoothing.

A smooth approximation to $\mathbf{I}^y(x) = \mathbf{I}^{(l)}(x,y)$ is then

$$\mathbf{S}^{(l)}(x,y) = \sum_{j=-n_x}^{n_x} \hat{\mathbf{I}}_j^y \hat{k}_j \, e^{j2\pi i x}, \qquad (22)$$

and its derivative with respect to x is

$$\frac{\partial \mathbf{S}^{(l)}}{\partial x} = 2\pi i \sum_{j=-n_x}^{n_x} j \hat{\mathbf{I}}_j^y \hat{k}_j \, e^{j2\pi i x}. \qquad (23)$$

Similarly, one can compute a smooth interpolant $\mathbf{S}^{(r)}$ for $\mathbf{I}^{(r)}$.

For $\mathbf{F}(\mathbf{d})(x,y) = \mathbf{F}^{(c)}(\mathbf{d})(x,y)$, the regularized least squares cost functional is then taken to be

$$\begin{aligned} f(\mathbf{d}) &= \frac{\alpha^2}{2} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} [\mathbf{S}^{(l)}(x_i - \frac{1}{2}\mathbf{d}(x_i, y_j), y_j) \\ &- \mathbf{S}^{(r)}(x_i + \frac{1}{2}\mathbf{d}(x_i, y_j), y_j)]^2 \\ &+ \int_\Omega \int \sqrt{\|\nabla \mathbf{d}(x,y)\|^2 + \beta^2} \, d\Omega. \end{aligned} \qquad (24)$$

To find a (local) minimizer, we apply the quasi-Newton iteration

$$\mathbf{d}^{m+1} = \mathbf{d}^m - H(\mathbf{d}^m)^{-1} \nabla(\mathbf{d}^m), \quad m = 0, 1, \ldots. \qquad (25)$$

The gradient $\nabla(\mathbf{d}^m)$ is computed exactly. The approximate Hessian takes the form

$$\mathbf{H}(\mathbf{d}) = \alpha^2 \mathbf{H}_{ls} + \mathbf{H}_\beta.$$

Here $\mathbf{H}_{ls}$ is a $n_x n_y \times n_x n_y$ diagonal matrix whose diagonal entry corresponding to pixel (x_i, y_j) is

$$\frac{1}{4}\left[\frac{\partial \mathbf{S}^{(l)}}{\partial x}(x_i - \frac{1}{2}\mathbf{d}(x_i, y_j), y_j) + \frac{\partial \mathbf{S}^{(r)}}{\partial x}(x_i + \frac{1}{2}\mathbf{d}(x_i, y_j), y_j)\right]^2.$$

This is obtained from the Gauss-Newton approximation [4] to the Hessian of the least squares term in (24). (Note from equation (13) that this approximation differs from the true Hessian by a term corresponding to $(\mathbf{F}(\mathbf{d}))\frac{\partial \mathbf{F}^2}{\partial \mathbf{d}^2}(\mathbf{d})$).

The matrix $\mathbf{H}_\beta$ is obtained from discretization of the diffusion operator $\mathbf{L}(\mathbf{d})$, cf., (15). The exact (discretized) Hessian takes the form $\mathbf{L}(\mathbf{d}) + \frac{\partial \mathbf{L}}{\partial \mathbf{d}}(\mathbf{d})$. This approximation yields the "lagged diffusivity fixed point iteration" in [12]. With lexicographical ordering of the unknowns, $\mathbf{H}_\beta$ is block tridiagonal with tridiagonal blocks. Since $\mathbf{H}_{ls}$ is diagonal, the approximate Hessian is also block tridiagonal with tridiagonal blocks. Hence, sparse matrix techniques can be used to efficiently invert the Hessian. While the asymptotic convergence rate for the quasi-Newton iteration (25) is only linear, in practice we have found convergence to be quite rapid.

Acknowledgement: Data for this research were partially provided by the Calibrated Imaging Laboratory at Carnegie Mellon University, supported by ARPA, NSF, and NASA. This research was parially supported by the Air Force Office of Scientific

Research AF/F49620-94-1-00S8DEF and AF/F49620-949190461, by the Army Research Office DAAL03-92-G-0115, DAAH04-94-G-0054, DAAH04-93-G-0332, and MURI Grant.

References

[1] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel, "Axioms and fundamental equations of image processing," *Arch. Rational Mechanics* **123** (1993), pp. 200-257.

[2] P. Burt and B. Julesz, "Modifications of the classical notion of Panum's fusional area," *Perception* **9** (1980), pp. 671-682.

[3] M. G. Crandall, H. Ishii, and P. L. Lions, "User's guide to viscosity solutions of second order partial linear differential equations," *Bulletin of the American Math. Society* **27** (1992), pp. 1-67.

[4] J. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, 1983.

[5] D. Dobson and F. Santosa, "Recovery of blocky images from noisy and blurred data", *SIAM Journal on Applied Math*, **56** (1996), pp. 1181-1198.

[6] U. R. Dhond and J. K Aggarwal, "Structure from stereo - A review," *IEEE Trans. Systems, Man, and Cybernetics* **19** (1989), pp. 1489 - 1510.

[7] Y. G. Chen, Y. Giga, and S. Goto, "Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations," *J. Differential Geometry* **33**, pp. 749-786, 1991.

[8] E. Giusti, *Minimal Surfaces and Functions of Bounded Variation*, Birkhäuser, Boston, 1984.

[9] B. K. P. Horn, *Robot Vision*, MIT Press, Cambridge, Mass., 1986.

[10] I. P. Howard and B. J. Rogers, *Binocular Vision and Stereopsis*, Oxford Univ. Press, New York, 1995.

[11] A. Kumar, A. Tannenbaum, and G. Balas, "Optical flow: a curve evolution approach," *IEEE Transactions on Image Processing* **5** (1996), pp. 598-610.

[12] M. Oman and C. Vogel, "Iterative methods for total variation denoising," *SIAM Journal on Scientific Computing* **17** (1996), pp. 227-238.

[13] S. Osher and L. I. Rudin, "Feature-oriented image enhancement using shock filters," *SIAM J. Numer. Anal.* **27** (1990), pp. 919-940.

[14] S. J. Osher and J. A. Sethian, "Fronts propagation with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics* **79** (1988), pp. 12-49.

[15] S. B. Pollard, J. E. Mayhew, and J. P. Frisby, "PMF: A stereo correspondence algorithm using a disparity gradient limit," *Perception* **14** (1981), pp. 449-470.

[16] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D* **60**, pp. 259-268, 1992.